

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(11)特許出願公開番号

(43)公開日 平成10年(1998)2月20日

技術表示箇所

3 2 2 B

(74)代理人 弁理士 後藤 洋介 (外2名)

```

graph TD
    1[入力装置] --> 21[構文図描画手段]
    21 --> 22[描画規則]
    21 --> 23[構文図描画制御部]
    22 --> 23
    23 --> 5[表示装置]
    23 --> 3
    subgraph 3 [構文図情報ファイル]
        direction LR
        3_1[(構文図情報ファイル)]
        3_2[(...)]
    end
    3 --> 41
    45[コードバッファ] --> 41
    44[構文図／コード変換部] --> 41
    43[構文図読解順序キュー] --> 41
    42[構文図読解情報ハッシュ] --> 41
    41 --> 6[BNF形式構文解析生成系]
    41 --> 7[字句解析生成系]
    6 --> 41
    7 --> 41
    41 --> CS1[Cソースコード]
    41 --> CS2[Cソースコード]
    CS1 --> 8[コンパイラ]
    CS2 --> 8
    8 --> 9[実行ファイル出力(コンパイラ)]
  
```

【特許請求の範囲】

【請求項1】 ノードとアークとを有する構文図の作成に必要な文字列及び画像データを入力情報として入力するための入力手段と、

該入力情報に基づいて前記構文図を描画して、複数のノード情報及び複数のアーク情報とを含む構文図情報ファイルを出力すると共に、前記構文図が構文図の描画規則である構文図描画規則に合致しない場合には、当該構文図を前記構文図描画規則に合致させるようにガイドするガイド情報を出力するための構文図情報生成手段と、該構文図情報ファイルを受けて、BNF形式構文解析生成系コード及び字句解析生成系コードに変換するためのコード生成手段と、前記ガイド情報を表示するための表示手段とを備えており、構文図をビジュアル言語として扱うことを特徴とするコンパイラ開発装置。

【請求項2】 請求項1に記載のコンパイラ開発装置において、

複数の前記構文図情報ファイルを記憶するための記憶手段を更に備えており、

前記コード生成手段は、前記記憶手段の記憶する前記複数の構文図情報ファイルを受けて、BNF形式構文解析生成系コード及び字句解析生成系コードに変換することを特徴とするコンパイラ開発装置。

【請求項3】 請求項2に記載のコンパイラ開発装置において、

前記構文図情報生成手段は、

前記入力情報に基づいて前記構文図を作成するための構文図描画手段と、

当該構文図が、前記構文図描画規則に合致するかどうかを検査するための描画規則検査手段と、

前記構文図描画手段及び前記描画規則検査手段からの出力を受けて、当該構文図が前記構文図描画規則に合致する場合には、当該構文図を前記構文図情報ファイルとして出力し、当該構文図が前記構文図描画規則に合致しない場合には、前記ガイド情報を前記表示手段に送出するための構文図描画制御部とを備えることを特徴とするコンパイラ開発装置。

【請求項4】 請求項2又は3のいずれかに記載のコンパイラ開発装置において、

前記コード生成手段は、

当該構文図情報ファイルの名称を格納するための構文図展開確認ハッシュと、

当該構文図情報ファイルに含まれる前記複数のノード情報であって、前記構文図展開確認ハッシュに格納されている構文図情報ファイルの名称に対応しない前記ノード情報である未展開ノード情報を格納するための構文図展開順序キューと、

前記構文図情報ファイルをBNF形式構文解析生成系コ

ード及び字句解析生成系コードに変換するための構文図／コード変換部と、

構文図／コード変換部において変換されたコードを一時的に格納するためのコードバッファと、

前記記憶手段から前記複数の構文図情報ファイルを順次読み出して、前記未展開ノード情報を前記構文図展開順序キューに格納させ、該構文図情報ファイルの名称を前記構文図展開確認ハッシュに格納させ、該構文図展開順序キューに格納された未展開ノード情報及び前記構文図展開確認ハッシュに格納された前記構文図情報ファイルの名称を参照して、前記構文図／コード変換部に対して構文図情報ファイルからコードへの変換指示をするためのコード生成制御部とを備えていることを特徴とするコンパイラ開発装置。

【請求項5】 請求項2乃至請求項4のいずれかに記載のコンパイラ開発装置において、

前記コード生成手段から前記BNF形式構文解析生成系コードを受けて第1の中間データを生成するためのBNF形式構文解析生成系と、

前記コード生成手段から前記字句解析生成系コードを受けて第2の中間データを生成するための字句解析生成系と、

前記第1及び第2の中間データを受けて、実行ファイルである所望のコンパイラに変換して該所望のコンパイラを出力するための変換部とを更に備えており、

前記構文図の描画から実行ファイルであるコンパイラを出力するまでを一貫して行えることを特徴とするコンパイラ開発装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンパイラを開発する際に描く構文図をBNF(Backus Naur form)形式構文解析生成系コード及び字句解析生成系コードへ変換する技術に関し、特に、階層化された構文図をBNF形式構文解析生成系コード及び字句解析生成系コードへ変換する装置及び方法に関する。

【0002】

【従来の技術】従来、コンパイラを開発する場合、コンパイラの仕様作成から実行ファイル(即ち、コンパイラ)の生成までの工程としては、例えば以下に示すようなものがあり、各工程は、夫々分離したものであった。

【0003】まず、第1の工程として、コンパイラの仕様を構文図に描いて作成する。次に、第2の工程として、その仕様を基にBNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(lexコード)のコーディングを行う。次に、第3の工程として、該コーディングしたコードをBNF形式構文解析生成系(例えば、yacc)及び字句解析生成系(lex)によりC言語へ変換しCソースコードを作成する。最後に、第4の工程として、作成したCソースコ

ードをCコンパイラにより実行ファイル(コンパイラ)へ変換する。

【0004】ここで、構文図をBNF形式構文解析生成系コード及び字句解析生成系コードへ変換する方式としては、図5に示される様に、“順次”、“選択”、“繰り返し”、及び“階層”といった基本単位の変換方式について規定されている。また、ここで用いられる構文図を作成する場合、非終端記号と終端記号の区別は、必ず明示されていなければならない。

【0005】また、構文図によって描かれたコンパイラの仕様を参照しながら、BNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)、並びにC言語などの高水準言語をコーディングする場合、コンパイラを開発する開発者が、テキストエディタを使用してコーディングするといった開発スタイルが主流であった。

【0006】このようなテキストエディタによる開発では、まず、BNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)をコーディングする。この際、コーディングしたものに、少しのバグ及び記述ミスがあっても問題はない。次に、該コーディングしたコードを、BNF形式構文解析生成系(例えば、yacc)及び字句解析生成系(例えば、lex)にかけて、Cソースコードなどへ変換させる。その際に、前述した多少のバグなどを有していると、エラーメッセージなどが示されるため、該エラーメッセージをもとにデバッグを行いながら完成させていた。

【0007】

【発明が解決しようとする課題】しかしながら、従来のコンパイラの開発過程においては、下記のような問題点を有していた。

【0008】従来のコンパイラ開発工程において、前記のようにコンパイラの仕様である構文図を作成する工程とは別に、該仕様を参照しながらコンパイラ開発者が手入力によりテキストエディタ等を用いて、C言語などの高水準言語、BNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)をコーディングする工程を必要とした。即ち、一旦、コンパイラの仕様を作成したにもかかわらず、更にコード化をしなければならないため、手間数が多く非効率的であるという問題を有していた。これは、構文図によって描かれたコンパイラの仕様をビジュアル言語としてではなく、単なる仕様としてしか捉えていなかったためである。

【0009】また、従来のようなテキストエディタによる開発においては、BNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)がバグを持っている場合に、バグの早期発見ができないという問題を有していた。こ

れは、コーディングしたコードがバグを有していることは、一般に起こり得ることであるにもかかわらず、従来の開発工程においては、BNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)を一旦作成終了(完成)した後でないと、該BNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)に対して、BNF形式構文解析生成系(例えば、yacc)及び字句解析生成系(例えば、lex)によるCソースコードへの変換命令を発効することができないためである。従って、当然のことながら、BNF形式構文解析生成系(例えば、yacc)及び字句解析生成系(例えば、lex)によるCソースコードへの変換命令を発効するまでは、該BNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)の記述ミスを発見することができなかった。

【0010】また、複数の構文図に同じノード名が繰り返し使用されている多重階層の構文図をBNF形式構文解析生成系コード及び字句解析生成系コードへ変換する際には、コーディングを行う開発者が逐次判断をしながら変換するという不便なものであった。これは、構文図をBNF形式構文解析生成系コード及び字句解析生成系コードへ変換する方式としては、前述した様に、基本単位の変換方式については、既に知られているが、多重階層の構文図についての変換方式が確立されておらず、従って、多重階層の構文図についての変換方式は提供されていなかったためである。

【0011】更に、従来、仕様としての構文図を作成する際には、前述した様に、必ず終端記号を明示しなければならない等、構文図を作成するのに時間がかかるという問題を有していた。これは、終端記号を自動的に判断できなかったためである。

【0012】本発明は、上述した種々の問題点を解決することを目的とする。

【0013】即ち、本発明の目的は、構文図によって描かれたコンパイラの仕様をBNF形式構文解析生成系コード及び字句解析生成系コードへ自動変換する装置を提供することにある。

【0014】また、本発明の他の目的は、構文図を描画する際に、逐次正しく描かれているかを検査して、誤りがあった場合には、該誤りを正す方向へガイドする機能を備えた装置を提供することにある。

【0015】また、本発明の他の目的は、多重階層の構文図をBNF形式構文解析生成系コード及び字句解析生成系コードへ変換する方式を提供することにある。

【0016】また、本発明の他の目的は、仕様である構文図を作成する際に、終端記号及び非終端記号の区別を明示しなくても終端記号を自動的に判断する機能を備えた装置を提供することにある。

【0017】

【課題を解決するための手段】本発明は、上述した課題を解決すべく、以下に示す手段を提供する。

【0018】即ち、本発明によれば、ノードとアークとを有する構文図の作成に必要な文字列及び画像データを入力情報として入力するための入力手段と、該入力情報に基づいて前記構文図を描画して、複数のノード情報及び複数のアーク情報とを含む構文図情報ファイルを出力すると共に、前記構文図が構文図の描画規則である構文図描画規則に合致しない場合には、当該構文図を前記構文図描画規則に合致させるようにガイドするガイド情報を出力するための構文図情報生成手段と、該構文図情報ファイルを受けて、BNF形式構文解析生成系コード及び字句解析生成系コードに変換するためのコード生成手段と、前記ガイド情報を表示するための表示手段とを備えており、構文図をビジュアル言語として扱うことを特徴とするコンパイラ開発装置が得られる。

【0019】また、本発明によれば、前記コンパイラ開発装置において、複数の前記構文図情報ファイルを記憶するための記憶手段を更に備えており、前記コード生成手段は、前記記憶手段の記憶する前記複数の構文図情報ファイルを受けて、BNF形式構文解析生成系コード及び字句解析生成系コードに変換することを特徴とするコンパイラ開発装置が得られる。

【0020】また、本発明によれば、前記コンパイラ開発装置において、前記構文図情報生成手段は、前記入力情報に基づいて前記構文図を作成するための構文図描画手段と、当該構文図が、前記構文図描画規則に合致するかどうかを検査するための描画規則検査手段と、前記構文図描画手段及び前記描画規則検査手段からの出力を受けて、当該構文図が前記構文図描画規則に合致する場合には、当該構文図を前記構文図情報ファイルとして出力し、当該構文図が前記構文図描画規則に合致しない場合には、前記ガイド情報を前記表示手段に送出するための構文図描画制御部とを備えることを特徴とするコンパイラ開発装置が得られる。

【0021】また、本発明によれば、前記コンパイラ開発装置において、前記コード生成手段は、当該構文図情報ファイルの名称を格納するための構文図展開確認ハッシュと、当該構文図情報ファイルに含まれる前記複数のノード情報であって、前記構文図展開確認ハッシュに格納されている構文図情報ファイルの名称に対応しない前記ノード情報である未展開ノード情報を格納するための構文図展開順序キューと、前記構文図情報ファイルをBNF形式構文解析生成系コード及び字句解析生成系コードに変換するための構文図／コード変換部と、構文図／コード変換部において変換されたコードを一時的に格納するためのコードバッファと、前記記憶手段から前記複数の構文図情報ファイルを順次読み出して、前記未展開ノード情報を前記構文図展開順序キューに格納させ、該

構文図情報ファイルの名称を前記構文図展開確認ハッシュに格納させ、該構文図展開順序キューに格納された未展開ノード情報及び前記構文図展開確認ハッシュに格納された前記構文図情報ファイルの名称を参照して、前記構文図／コード変換部に対して構文図情報ファイルからコードへの変換指示をするためのコード生成制御部とを備えていることを特徴とするコンパイラ開発装置が得られる。

【0022】更に、本発明によれば、前記コンパイラ開発装置において、前記コード生成手段から前記BNF形式構文解析生成系コードを受けて第1の中間データを生成するためのBNF形式構文解析生成系と、前記コード生成手段から前記字句解析生成系コードを受けて第2の中間データを生成するための字句解析生成系と、前記第1及び第2の中間データを受けて、実行ファイルである所望のコンパイラに変換して該所望のコンパイラを出力するための変換部とを更に備えており、前記構文図の描画から実行ファイルであるコンパイラを出力するまでを一貫して行えることを特徴とするコンパイラ開発装置が得られる。

【0023】

【発明の実施の形態】以下に、本発明の実施の形態のコンパイル装置について図面を参照して説明する。

【0024】本実施の形態のコンパイラ開発装置は、図1に示されるような構成を備えている。

【0025】即ち、本実施の形態のコンパイラ開発装置は、ノードとアークとを有する構文図の作成に必要な文字列及び画像データである情報を入力する入力装置1と、入力装置1に入力された情報に基づいて構文図を描画して構文図情報ファイルを出力する構文図情報生成手段2と、構文図情報生成手段2の出力を受けて複数の構文図情報ファイルを記憶するファイルバッファ3と、ファイルバッファ3に記憶された複数の構文図情報ファイルからBNF形式構文解析生成系コード（例えば、yaccコード）及び字句解析生成系コード（例えば、lexコード）を生成するコード生成手段4と、表示手段5と、BNF形式構文解析生成系コードを第1の中間データ（本実施の形態においては、Cソースコード；以下、同じ。）に変換するBNF形式構文解析生成系6と、字句解析生成系コードを第2のCソースコードに変換する字句解析生成系7と、BNF形式構文解析生成系6と字句解析生成系7の夫々の出力したCソースコードを受けてコンパイルして実行データ（新しく開発したコンパイラ）を出力するCコンパイラ8とを備えている。

【0026】更に詳しくは、構文図情報生成手段2は、構文図描画手段21と、描画規則検査手段22と、構文図描画制御部23とを備えており、構文図描画手段21において、入力装置1に入力された文字列及び画像データ等の情報に従い、構文図を描画し、描画規則検査手段22において、描画した構文図が構文図描画規則（構文

図の描画規則)に合致するかどうかを検査し、構文図描画制御部23において、構文図描画手段21と描画規則検査手段22からの出力を受けて、当該構文図が構文図描画規則に合致する場合には、当該構文図を構文図情報ファイルとしてファイルバッファ3に出力し、当該構文図が構文図描画規則に合致しない場合には、当該構文図を構文図描画規則に合致させるようにガイドするガイド情報を表示装置5に送出する。

【0027】また、コード生成部4は、コード生成制御部41と、構文図展開確認ハッシュ42と、構文図展開順序キュー43と、構文図／コード変換部44と、コードバッファ45とを備えており、入力装置1からコード生成制御部41が、ファイルバッファ3から構文図情報ファイルを順次読み出して、未展開ノード情報を構文図展開順序キュー43に格納し、該構文図情報ファイルを構文図／コード変換部44においてBNF形式構文解析生成系コード及び字句解析生成系コードにコード変換させて、展開した構文図情報ファイルの名称を構文図展開確認ハッシュ42に格納すると共に、変換されたコードをコードバッファ45に格納する。

【0028】ここで、未展開ノード情報とは、構文図情報ファイルに含まれる複数のノード情報であって、構文図展開確認ハッシュ42に格納されている構文図情報ファイルの名称に対応しないノード情報のことである。

【0029】尚、本実施の形態で用いられる構文図は、基本的には一般の構文図と同じであるが、一部、制約及び拡張した部分があるので以下に説明する。

【0030】本実施の形態においては、構文図を単純化するために、図2に示される(a)通常記述構文図と

(b)再起記述構文図の2種類に限定することとし、また、混在は認めないこととする。また、データ項目は、開始ノード、終了ノード、非終端ノード及び終端ノードからなる。ここで、非終端ノード及び終端ノードは、見掛上区別がないものとする。また、構文図には開始ノード及び終了ノードが予め設定されている。各構文図において、矢印の始まり部分が開始ノードであり、矢印の末端部分が終了ノードである。各項目の連なりは構文図と同様に矢つき線(アーク)で示され、左から右にアークにそって読み進むことにより並びの規則を把握することができる。尚、階層的な記法は用意されていないので、「繰返し」を表現するには、再起記述構文図を用いる。

【0031】次に、このような構成を備えたコンパイラ開発装置におけるコード生成の動作を説明する。

【0032】まず、図3に示されるように、入力装置1から構文図作成に必要な文字列及び画像データを入力情報として入力する(ステップS101)。次に、該入力情報に従い、構文図描画手段21において構文図を描画する(ステップS102)。次に、該描画した構文図が

構文図描画規則に合致しているかどうか描画規則検査手段22において検査する(ステップS103)。ステップS103において、該構文図が構文図描画規則に合致していないと判断した場合、構文図描画制御部23は、該構文図を構文図描画規則に合致させるようにガイドするガイド情報を表示装置5に表示させる(ステップS104)。一方、ステップS103において、該構文図が構文図描画規則に合致していると判断した場合、構文図描画制御部23は、該構文図を構文図情報ファイルとしてファイルバッファ3に格納する(ステップS105)。ここで、全ての構文図を描画し終えたかどうかを判断し(ステップS106)、まだ描画しなければならない構文図がある場合は、再びステップS101へ戻り構文図の描画を続行する。一方、ステップS106においてコンパイラ生成に必要な全ての構文図を描画し終えたと判断した場合、入力装置1からコード生成制御部41へコード変換指示を発効する(ステップS107)。

【0033】次に、コード変換指示を受けたコード生成部4側の処理について図4を用いて説明する。

【0034】まず、コード生成制御部41は、コード変換指示を受けると、構文図展開確認ハッシュ42、構文図展開順序キュー43、及びコードバッファ45等の初期化を行う(ステップS201)。ここで、構文図展開順序キュー43の初期値は、ルート構文図である構文図の名称を示すものとし、また構文図展開確認ハッシュ42には何も格納されていないものとする。次に、コード生成制御部41は、構文図展開順序キュー43を参照して、ファイルバッファ3から該当する構文図情報ファイルを読み出す(ステップS202)。コード生成制御部41は、構文図情報ファイルを読み込むと、該構文図情報ファイルに含まれる未展開ノードを構文図展開順序キューに格納する(ステップS203)。ここで、未展開ノードとは、前述の通り、該構文図情報ファイルに含まれるノードであって、構文図展開確認ハッシュ42に格納されている構文図情報ファイルの名称に対応しないノードである。従って、最初のステップS203においては読み込んだ構文図情報ファイルに含まれる全てのノードが格納されることになる。次に、構文図／コード変換部44において、当該構文図情報ファイルをBNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)に変換し(ステップS204)、変換して得られたBNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)をコードバッファ45へ格納する(ステップS206)。次に、コード生成制御部41は、構文図展開順序キュー43が空であるかどうかを判定し(ステップS207)、構文図展開順序キュー43が空でなければ、構文図展開順序キュー43に格納されている未展開ノードを参照して、次の構文図情報ファイルをファイルバッファ3から

読み込む(ステップS208)。次に、ステップS208において、該当する名称の構文図情報ファイルを読み込んだかどうかを判定し(ステップS209)、読み込んだ場合は再びステップS203へ戻り、処理を続行する。ステップS209において、該当する名称の構文図情報ファイルを読み込めないと判定した場合は、ステップS207へ戻り構文図展開順序キュー43が空になったかどうかを判定して、処理を続行する。ステップS207において、構文図展開順序キュー43が空であると判定した場合、ファイルバッファ3に、まだ読み込んでない構文図情報ファイルがあるかどうか判定し(ステップS210)、読み込んでない構文図情報ファイルがある場合は、表示装置1へガイド情報を表示する(ステップS211)。ステップS210において、読み込んでない構文図情報ファイルがないと判定した場合、コードバッファ45からBNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)を、夫々、BNF形式構文解析生成系(例えば、yacc)6及び字句解析生成系(例えば、lex)7に出力する(ステップS212)。このようにして、得られたBNF形式構文解析生成系コード(例えば、yaccコード)及び字句解析生成系コード(例えば、lexコード)を、夫々、BNF形式構文解析生成系(例えば、yacc)6及び字句解析生成系(例えば、lex)7において、Cソースコードに変換して、該CソースコードをCコンパイラにより、コンパイルし、実行ファイル(開発したコンパイラ)を得る。尚、本実施の形態においては、一例として、ステップS210で、読み込んでない構文図情報ファイルがあるかないかだけを検査していたが、同ステップSを階層構文図全般の規則に合致しているかどうかを検査するものとしても良い。

【0035】

【発明の効果】本発明によれば、構文図をビジュアル言語として扱うことが出来、さらにガイド機能を備えていることから早期にエラーを発見することが出来るため、コンパイラの開発効率の向上を図ることが出来るコンパイラ開発装置が得られる。

【0036】また、本発明によれば、終端記号及び非終

端記号の区別を明示しなくても良く、更に多重階層の構文図をBNF形式構文解析生成系コード及び字句解析生成系コードへ変換することが出来ることから、コンパイラ生成系固有の文法を習熟する必要がないため、従来より簡単にコンパイラを開発することが出来るコンパイラ開発装置が得られる。

【0037】更に、本発明のコンパイラ開発装置において描画された構文図は、そのままドキュメントとして利用可能であることからドキュメントとしての保守性の向上が図れる。

【図面の簡単な説明】

【図1】本発明の実施の形態のコンパイラ開発装置の構成を示す機能ブロック図である。

【図2】本発明の実施の形態において用いる構文図の説明のための図である。

【図3】本発明の実施の形態において、構文図描画からコード変換指示を発効するまでの動作を説明するフローチャートである。

【図4】本発明の実施の形態において、コード変換指示を発効した後の動作を説明するフローチャートである。

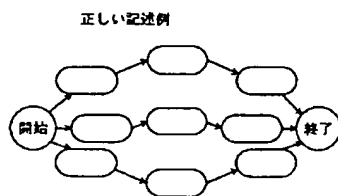
【図5】従来の構文図をBNF記法へ変換する方法を示す図である。

【符号の説明】

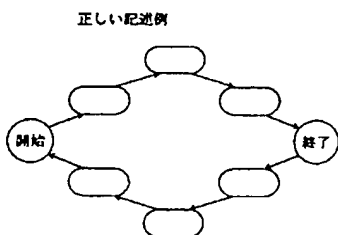
- | | |
|----|--------------|
| 1 | 入力装置 |
| 2 | 構文図情報生成手段 |
| 21 | 構文図描画手段 |
| 22 | 描画規則検査手段 |
| 23 | 構文図描画制御部 |
| 3 | ファイルバッファ |
| 4 | コード生成部 |
| 41 | コード生成制御部 |
| 42 | 構文図展開確認ハッシュ |
| 43 | 構文図展開順序キュー |
| 44 | 構文図／コード変換部 |
| 45 | コードバッファ |
| 5 | 表示装置 |
| 6 | BNF形式構文解析生成系 |
| 7 | 字句解析生成系 |
| 8 | Cコンパイラ |

[illegible]

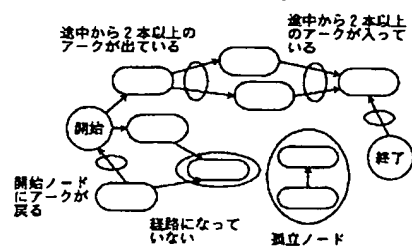
(a) 通常記述構文図



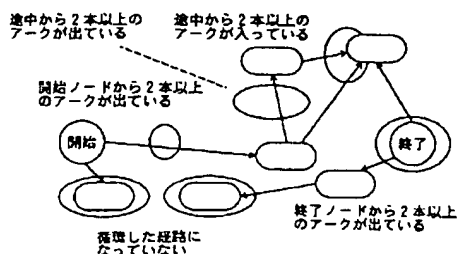
(b) 再編記述構文図



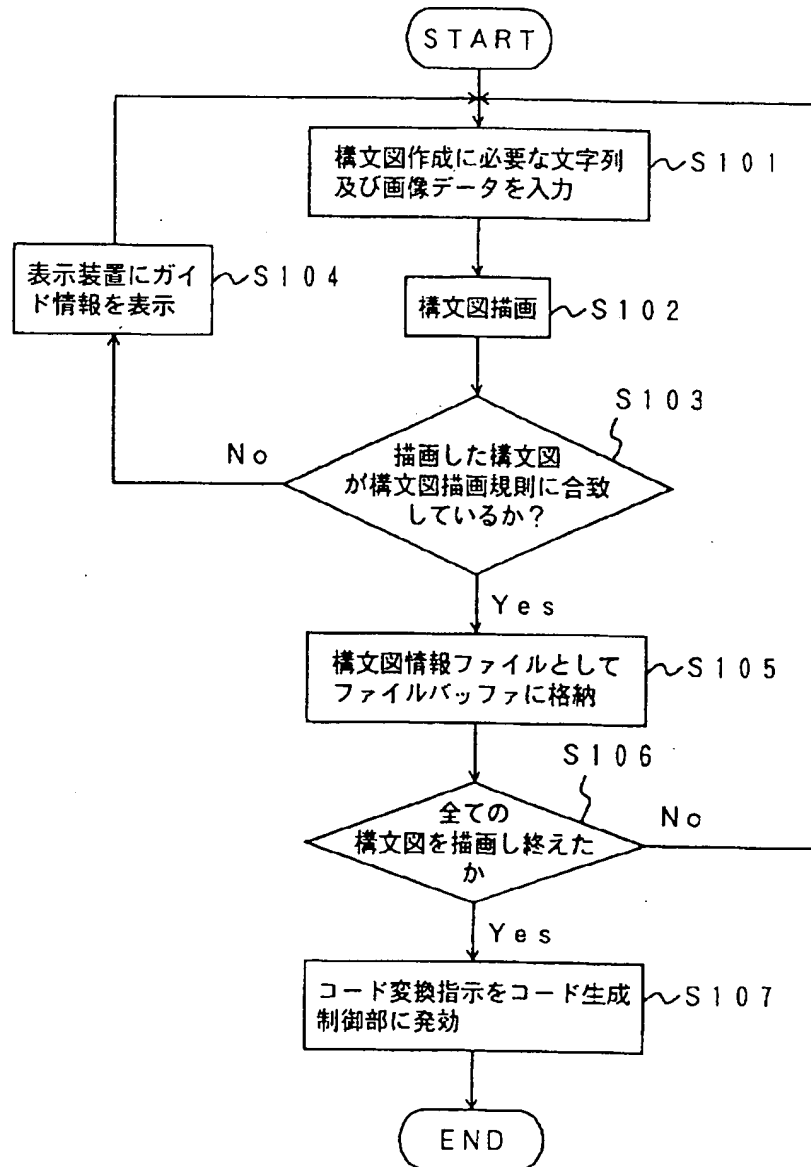
誤った記述例



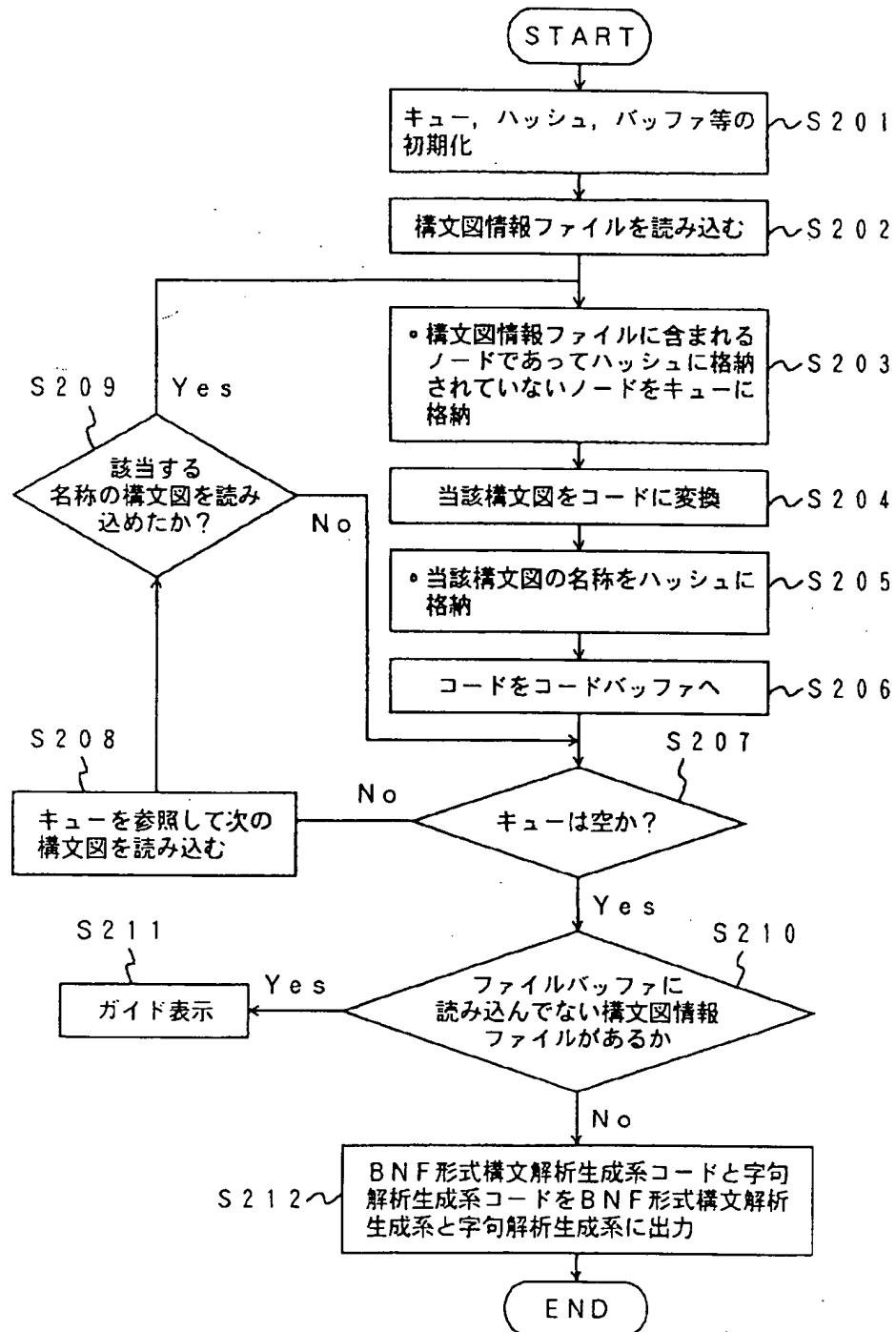
缺った記述例



【図3】



【図4】



【図5】

